UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/847,063	04/30/2001	Ming Zhou	GEI-001US 29083	4555

21718 7590 04/16/2004	EXAMINER
LEE & HAYES PLLC	RUTTEN, JAMES D

SUITE 500	ART UNIT	PAPER NUMBER
421 W RIVERSIDE	2122	
SPOKANE, WA 99201		

DATE MAILED: 04/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

	Application No.	Applicant(s)
Office Action Summary	09/847,063	ZHOU ET AL.
	Examiner	Art Unit
	J. Derek Rutten	2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) FROM
THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _30 April 2001_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is
closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-56_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-56_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on _30 April 2001_ is/are: a)☐ accepted or b)☒ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☒ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All b)☐ Some * c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage

        application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

1.   Claims 1-56 have been examined.

Oath/Declaration

2.   The oath or declaration is defective. A new oath or declaration in compliance with 37

CFR 1.67(a) identifying this application by application number and filing date is required. See

MPEP §§ 602.01 and 602.02.

> The oath or declaration is defective because:
> While it acknowledges the duty to disclose as defined in 37 CFR **1.56(a)**, it does not state
> that the person making the oath or declaration acknowledges the duty to disclose to the
> Office all information known to the person to be material to patentability as defined in 37
> CFR **1.56**.

Drawings

3.   Figures 1-3 should be designated by a legend such as --Prior Art-- because only that

which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected

drawings are required in reply to the Office action to avoid abandonment of the application. The

objection to the drawings will not be held in abeyance.

Specification

4.   The use of the trademark "JSP" has been noted in this application on pages 3, 18, 19, and

23 and JavaScript on page 19. It should be capitalized wherever it appears and be accompanied

by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary

nature of the marks should be respected and every effort made to prevent their use in any manner

which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 102

5.　　The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6.　　Claims 1-5, 13-15, 18-24, 26-36, 45-47, 49, 51-54, and 56 are rejected under 35

U.S.C. 102(b) as being anticipated by U.S. Patent 5,678,039 to Hinks et al. (hereinafter referred

to as "Hinks").

As per claim 1, Hinks discloses:

A computer-implemented method comprising:

analyzing a computer-servable document written for a particular locale (column

3 lines 7-9: "First, the sources are **parsed** by the Export/Import

module to a translatable format."); *and*

extracting locale-sensitive content from the document while leaving locale-

independent elements in the document (column 3 lines 1-4: "The Export/Import

module itself includes a parsing engine to **extract** strings

and translatable information from application programs."; also

column 3 lines 39-43: "In either instance, the underlying program

code (i.e., the code which the programmer has written to

carry out the functionality of the program) has remained

untouched by the process.").

As per claim 2, the above rejection of claim 1 is incorporated. Further, Hinks

discloses *wherein the analyzing comprises examining each line of code in the document*

and based on the code, identifying the locale-sensitive content (column 3 lines 1-4).

As per claim 3 the above rejection of claim 1 is incorporated. Further, Hinks

discloses *wherein the locale-sensitive content comprises natural language text* (column 3

lines 28-31).

As per claim 4 the above rejection of claim 1 is incorporated. Further, Hinks

discloses *wherein the locale-independent elements comprise source code and formatting*

data (column 3 lines 31-34).

As per claim 5 the above rejection of claim 1 is incorporated. Further, Hinks

discloses *storing the locale-sensitive content in a data structure separate from the*

document (column 3 lines 13-16).

As per claim 13, Hinks discloses:

A method comprising:

automatically compiling a computer-servable document written for a particular

locale to extract any locale-sensitive content (column 3 lines 1-4, and 7-9 as cited in the

above rejection of claim 1), *the compiling producing a compiled document with locale-*

independent elements (FIG. 3 element 377 and column 8 lines 30-34:

"Alternatively, a Resource Compiler 365, again such as

Borland's Resource Workshop.RTM., may be employed to re-

compile the Translated Resource Files 360 and bind those

compiled resources back into the target program, now shown

as Translated Program 377."); *and*

storing the locale-sensitive content in a form that can be translated to other

locales (column 3 lines 15-22: "From there, Export/Import (EXPIMP)

module parses the resource file into a Translation Table,

which is typically **stored** as a database table. The

Translation Table encapsulates all the information that is

known or can be derived from the various resources and

stores them in a format which may be utilized by various

editors."; also column 3 lines 53-55: "In this fashion, changing a

product from one locale to another can be reduced to the

simple process of swapping out resource files.").

As per claims 14 and 15, all limitations have been addressed in the above

rejections of claims 3 and 4, respectively.

As per claim 18, the above rejection of claim 13 is incorporated. Further, Hinks

discloses *retrieving, at runtime, the compiled document and populating the compiled*

document with the locale-sensitive content to reconstruct the computer-servable

document that can be served to the particular locale (column 3 lines 50-55).

As per claim 19, the above rejection of claim 13 is incorporated. Further, Hinks

discloses *wherein the locale-sensitive content is translated in to a second version for use*

in a second locale (column 3 lines 35-36). All further limitations have been addressed in

the above rejection of claim 18.

As per claim 20, Hinks discloses:

compiling a computer-servable document written for a particular locale to extract

any locale-sensitive content, the compiling producing a compiled document with locale-

independent elements (column 3 lines 1-4, and 7-9 as cited in the above rejection of claim

1; also FIG. 3 element 377 and column 8 lines 30-34 as cited in the above rejection of

claim 13);

storing the locale-sensitive content (column 3 lines 15-22 as cited in the above

rejection of claim 13); *and*

at runtime, retrieving the compiled document and populating the compiled document with the locale-sensitive content (column 3 lines 50-55 as cited in the above rejection of claim 18).

As per claims 21 and 22, the above rejection of claim 20 is incorporated. All further limitations have been addressed in the above rejection of claims 3 and 4, respectively.

As per claim 23, the above rejection of claim 20 is incorporated. Further, Hinks discloses *storing the locale-sensitive content in a structured text file* (column 3 lines 9-11. Resource files structured text files.).

As per claim 24, the above rejection of claim 20 is incorporated. Further, Hinks discloses *storing the locale sensitive content in a database file* (column 3 lines 13-16).

As per claim 26, the above rejection of claim 26 is incorporated. Hinks further discloses:

storing one or more translated versions of the locale-sensitive content (column 3 lines 23-24); *and*

at runtime, retrieving the compiled document and populating the compiled document with a translated version of the locale-sensitive content (column 3 lines 50-55).

As per claim 27, Hinks discloses:

retrieving a locale-independent core that contains locale-independent elements (column 3 lines 1-4 as cited in the above rejection of claim 1); *and*

populating the locale-independent core with locale-sensitive content that is appropriate for the particular locale (column 3 lines 50-55 as cited in the above rejection of claim 18).

As per claims 28 and 29, the above rejection of claim 27 is incorporated. All other limitations have been addressed in the above rejections of claims 3 and 4, respectively.

As per claim 30, the above rejection of claim 27 is incorporated. Hinks further discloses *wherein the locale-independent core contains a function call in place of where corresponding locale-sensitive content reside in the document* (column 3 lines 36-43; Comment: Hinks source code makes function calls to access the resource files which contain the locale-sensitive content.), *and the populating comprises executing the function call to obtain the locale-sensitive content and insert the locale-sensitive content into the core* (column 3 lines 50-55).

As per claim 31, the above rejection of claim 27 is incorporated. Hinks further discloses *wherein multiple translations of the locale-sensitive content exist for multiple locales, and further comprising populating the locale-independent core with a translation*

of the locale-sensitive content, the translation being suitable for a corresponding locale

to which the document is to be served (column 3 lines 50-55).

As per claim 32, Hinks discloses a system (FIG. 3). Hinks further discloses:

at least one computer-servable document stored in a computer-readable medium,

the document being written for a particular locale (column 3 lines 7-9: "First, the

`sources` are parsed by the Export/Import module to a

translatable format."; also column 5 lines 59-61: "Software system

200, which is stored in system memory 102 and on `disk`

`memory` 107, includes a kernel or operating system (OS) 240

and a windows shell 250."); *and*

a compiler to automatically extract locale-sensitive content from the document to

produce a compiled document containing locale-independent elements (column 3 lines 7-

7: "First, the sources are parsed by the `Export/Import module`

to a translatable format."; also column 3 lines 50-51: "Resources for

products to be translated are stored in an `external`

`resource file` in a standard format.").

As per claims 33 and 34, the above rejection of claim 32 is incorporated. All

further limitations have been addressed in the above rejections of claims 3 and 4,

resepectively.

As per claim 35, the above rejection of claim 32 is incorporated. Hinks further discloses *wherein the compiler examines source code in the document to determine, from the source code, whether locale-sensitive content is present* (column 3 lines 7-9).

As per claim 36, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejection of claim 5.

As per claim 45, Hinks discloses:

A system (FIG. 3) *comprising:*

a resource bundle containing locale-specific content that is authored for a particular locale (column 3 lines 28-34: "The translated **resource file** is similar to the original resource file, except that any necessary **translations** (e.g., translating an English text string into a French text string) have been carried out."); *and*

a resource bundle manager to populate a locale-independent document core with the locale-sensitive content in the resource bundle to produce a computer-servable document that is suitable to be served to the particular locale (column 8 lines 14-16: "Once the end-user translator has completed the task of translating the resources, the translated text is merged back to sources."; also column 8 lines 30-34: "Alternatively, a Resource Compiler 365, again such as Borland's Resource

Workshop®, may be employed to re-compile the Translated

Resource Files 360 and bind those compiled resources back

into the target program, now shown as Translated Program

377.").

As per claim 46, the above rejection of claim 45 is incorporated. Hinks further

discloses *wherein the resource bundle is embodied as a database file* (column 3 lines 13-

16).

As per claim 47, the above rejection of claim 45 is incorporated. All other

limitations have been addressed in the above rejection of claim 3.

As per claim 49, Hinks discloses:

A system (FIG. 3) comprising:

compilation means for compiling a computer-servable document written for a

particular locale to extract any locale-sensitive content (column 3 lines 1-4, and 7-9 as

cited in the above rejection of claim 1), *the compilation means producing a compiled*

document with locale-independent elements (FIG. 3 element 377 and column 8 lines 30-

34 as cited in the above rejection of claim 13); *and*

storage means for storing the locale-sensitive content extracted from the

computer-servable document in a data structure separate from the compiled document

(column 3 lines 13-16 as cited in the above rejection of claim 5).

As per claim 51, the above rejection of claim 49 is incorporated. All further limitations have been addressed in the above rejection of claim 18.

As per claim 52, the above rejection of claim 49 is incorporated. All further limitations have been addressed in the above rejection of claim 19.

As per claim 53, Hinks discloses a system (FIG. 3). All further limitations have been addressed in the above rejection of claim 27.

As per claim 54, the above rejection of claim 53 is incorporated. All further limitations have been addressed in the above rejection of claim 31.

As per claim 56, Hinks discloses computer-readable media (column 5 line 44: "main memory"). All further limitations have been addressed in the above rejection of claim 27.

7.   Claims 7, 8, 10, and 12 are rejected under 35 U.S.C. 102(b) as being anticipated by "OpenWindows Developer's Guide: Xview Code Generator Programmer's Guide" by Sun Microsystems (hereinafter referred to as "Sun").

As per claim 7, Sun discloses:

A method comprising:

examining source code in a document written for a particular locale (page 97

"Text Databases": "To facilitate internationalization of text, a

process exists (see gettext(), later in this section) to

collect all text visible to the user into a file called a

portable object file. The portable object file contains

the **native language** strings from a program and placeholders

for a localizer to put each string's translation.");

determining, from the source code, locale-sensitive content that is specific to the

particular locale (page 97 as cited above, "native language strings");

extracting the locale-sensitive content from the source code (page 97 as cited

above, "collect all text");

storing the locale-sensitive content in a separate file (page 97 as cited above,

"portable object file"); *and*

inserting, into the document in place of the locale-sensitive content, a reference to

the locale-sensitive content in the separate file (page 98 under the header gettext() and

dgettext() Routines: "Two similar **routines** are available to a

developer for **retrieving** translated text. One, gettext(),

assumes a text domain has already been specified..."; page 99

under the header xgettext and msgfmt Utilities: "Once Devguide or a

developer has **inserted** gettext() function calls around all

user visible text in an application, xgettext can be run on

the source files to produce the portable object files.").

As per claim 8, the above rejection of claim 7 is incorporated. Sun further

discloses *wherein the locale-sensitive content comprises natural language text* (page 97

"native language strings").

As per claim 10, the above rejection of claim 7 is incorporated. Sun further

discloses *wherein the storing comprises storing the locale sensitive content in a*

structured text file (page 99 "portable object file").

As per claim 12, the above rejection of claim 7 is incorporated. Sun further

discloses *wherein the reference comprises a function call that, when executed, obtains the*

locale-sensitive content from the separate file (page 98 "retrieving translated text").

Claim Rejections - 35 USC § 103

8.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9.   Claims 6, 16, 17, 25, 37, 38, 41-44, 48, 50, and 55 are rejected under 35 U.S.C. 103(a) as

being unpatentable over Hinks as applied to claims 5, 13, 20, 36, 45, and 49, respectively above,

and further in view of Sun.

        As per claim 6, the above rejection of claim 5 is incorporated. Further, Hinks

does not expressly disclose the substitution of a locale-sensitive content with a function

call.

        However, in an analogous environment, Sun teaches *substituting a function call in*

place of associated locale-sensitive content in the document (page 99 under the header

"xgettext and msgfmt Utilities"), *the function call being configured such that, when*

executed, the function call obtains the locale-sensitive content from the data structure

(page 98 under the header "gettext() and dgettext() Routines").

        It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Sun's function call substitution in Hinks' localization system.

One of ordinary skill would have been motivated to include nothing specific to one's

language and culture in software development and to provide features that facilitate

translation of text into other languages.

        As per claim 16, the above rejection of claim 13 is incorporated. Further, Hinks

does not expressly disclose substituting a reference in place of the locale-sensitive

content in the compiled document.

However, in an analogous environment, Sun teaches *substituting a reference in place of the locale-sensitive content in the compiled document* (page 99 under the header "xgettext and msgfmt Utilities"), *the reference identifying the locale-sensitive content extracted from the compiled document* (page 98 under the header "gettext() and dgettext() Routines")

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference substitution in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 17, the above rejection of claim 13 is incorporated. Further, Hinks does not expressly disclose the substitution of a locale-sensitive content with a function call.

However, in an analogous environment, Sun teaches *substituting a function call in place of the locale-sensitive content in the compiled document* (page 99 under the header "xgettext and msgfmt Utilities"), *the function call being configured such that, when executed, the function call obtains and embeds the locale-sensitive content back into the compiled document* (page 98 under the header "gettext() and dgettext() Routines")

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function call substitution in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's

language and culture in software development and to provide features that facilitate

translation of text into other languages.

As per claim 25, the above rejection of claim 20 is incorporated. Further, Hinks

discloses obtaining the associated locale-sensitive content and inserting the associated

locale-sensitive content back into the compiled document (column 3 lines 50-55).

Hinks does not expressly disclose *substituting a function call in place of*

associated locale-sensitive content in the compiled document; and the populating

comprises executing the function call in the compiled document.

However, in an analogous environment, Sun teaches *substituting a function call in*

place of associated locale-sensitive content in the compiled document (page 99 under the

header "xgettext and msgfmt Utilities" as cited in the above rejection of claim 6) *and the*

populating comprises executing the function call in the compiled document (page 98

under the header "gettext() and dgettext() Routines" as cited in the above rejection of

claim 6).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Sun's function call substitution and population in Hinks'

localization system. One of ordinary skill would have been motivated to include nothing

specific to one's language and culture in software development and to provide features

that facilitate translation of text into other languages.

As per claim 37, the above rejection of claim 36 is incorporated. Further, Hinks does not expressly disclose inserting a reference in place of the locale-sensitive content in the document.

However, in an analogous environment, Sun teaches *inserting a reference in place of the locale-sensitive content in the compiled document* (page 99 under the header "xgettext and msgfmt Utilities"), *the reference identifying the locale-sensitive content in the data structure* (page 98 under the header "gettext() and dgettext() Routines").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference insertion in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 38, the above rejection of claim 36 is incorporated. Further, Hinks discloses obtaining the associated locale-sensitive content and inserting the associated locale-sensitive content back into the compiled document (column 3 lines 50-55).

Hinks does not expressly disclose *substituting a function call in place of associated locale-sensitive content in the compiled document.*

However, in an analogous environment, Sun teaches *substituting a function call in place of associated locale-sensitive content in the compiled document* (page 99 under the header "xgettext and msgfmt Utilities" as cited in the above rejection of claim 6) *the function call being configured such that, when executed, the function call obtains the*

associated locale-sensitive content from the data structure and inserts the associated

locale-sensitive content back into the compiled document (page 98 under the header

"gettext() and dgettext() Routines" as cited in the above rejection of claim 6).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Sun's function call substitution and population in Hinks'

localization system. One of ordinary skill would have been motivated to include nothing

specific to one's language and culture in software development and to provide features

that facilitate translation of text into other languages.

As per claim 41, Hinks discloses:

A compiler system (FIG. 3) *comprising:*

a grammar containing rules for structuring source code (column 3 lines 7-9 as

cited in the above rejection of claim 1 describes a parser which inherently uses a source

code grammar, otherwise it would be unable to match tokens of the source code.); *a*

content analyzer to analyze source code in a document written for a particular

locale and to utilize the grammar to determine whether the source code contains locale-

sensitive content that is specific to the particular locale (column 3 lines 7-9 as cited in the

above rejection of claim 1),

Hinks does not expressly disclose *the content analyzer being configured to*

replace the locale-sensitive content in the source code.

However, in an analogous environment, Sun teaches *the content analyzer being*

configured to replace the locale-sensitive content in the source code with associated

references to the replaced locale-sensitive content (page 99 under the header "xgettext and msgfmt Utilities", and page 98 under the header "gettext() and dgettext() Routines" as referenced in the above rejection of claim 6).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference replacement in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 42, the above rejection of claim 41 is incorporated. Hinks further discloses *wherein the locale-sensitive content is placed in a separate file,* (column 3 lines 13-16).

Hinks does not expressly disclose *and the references comprise a pointer to that file.*

However, Sun teaches a reference to a file containing locale-sensitive content (page 97 under "Text Databases (Text Domains)".

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference pointer with Hinks' content file. One of ordinary skill would have been motivated to retrieve text in the local language from a program with placeholders for a localizer to put each string's translation.

As per claim 43, the above rejection of claim 41 is incorporated. Hinks further

discloses *wherein the locale-sensitive content is placed in a separate data structure*

(column 3 lines 13-16 as cited in claim 5).

Hinks does not expressly disclose references that comprise function calls.

However, Sun teaches *the references comprise function calls that, when executed,*

obtain the associated locale-sensitive content from the data structure and insert the

associated locale-sensitive content into the source code (page 99 under the header

"xgettext and msgfmt Utilities" and page 98 under the header "gettext() and dgettext()

Routines").

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Sun's function calls to reference Hinks' data structure. One of

ordinary skill would have been motivated to retrieve text in the local language from a

program with placeholders for a localizer to put each string's translation.

As per claim 44, the above rejection of claim 41 is incorporated. Hicks further

discloses:

a call library to store function calls (column 5 lines 26-29 describe use of the

Microsoft Windows environment, which inherently contains a call library).

Hicks does not expressly disclose replacing locale-sensitive content with function

calls.

However, Sun teaches *the content analyzer replaces the locale-sensitive content*

with one or more function calls from the call library, the function calls being configured

such that, when executed, the function calls reinsert the locale-sensitive content back into the source code (page 99 under the header "xgettext and msgfmt Utilities" and page 98 under the header "gettext() and dgettext() Routines").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function calls to reference Hinks' data structure. One of ordinary skill would have been motivated to retrieve text in the local language from a program with placeholders for a localizer to put each string's translation.).

As per claim 48, the above rejection of claim 45 is incorporated. Hinks further discloses a core containing source code and function calls (column 3 lines 39-43).

Hinks does not expressly disclose execution of embedded function calls.

However, Sun teaches execution of embedded function calls to retrieve locale-specific content for inclusion in source code (page 99 under the header "xgettext and msgfmt Utilities" and page 98 under the header "gettext() and dgettext() Routines").

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's function calls in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 50, the above rejection of claim 49 is incorporated. Hinks does not expressly disclose substitution with a reference.

However, Sun teaches all further limitations as have been addressed in the above rejection of claim 16.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference substitution in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

As per claim 55, Hinks discloses:

One or more computer-readable media (column 5 line 44: "main memory") *comprising computer-executable instructions that, when executed, direct a computer to:*

examine source code in a document written for a particular locale (column 3 lines 7-9: "First, the sources are **parsed** by the Export/Import module to a translatable format.");

extract any locale-sensitive content from the source code (column 3 lines 1-4: "The Export/Import module itself includes a parsing engine to **extract** strings and translatable information from application programs.");

store the locale-sensitive content in a separate file (column 3 lines 13-16: "From there, Export/Import (EXPIMP) module parses the resource file into a Translation Table, which is typically stored as a database table.").

Hinks does not expressly disclose: *substitute, in place of the locale-sensitive content in the document, a reference to the locale-sensitive content in the separate file.*

However, Sun teaches: *substitute, in place of the locale-sensitive content in the document, a reference to the locale-sensitive content in the separate file* (page 99 under the header "xgettext and msgfmt Utilities" and page 98 under the header "gettext() and dgettext() Routines).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Sun's reference substitution in Hinks' localization system. One of ordinary skill would have been motivated to include nothing specific to one's language and culture in software development and to provide features that facilitate translation of text into other languages.

10.   Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sun as applied to claim 7 above, and further in view of U.S. Patent Application Publication US 2002/0107684 to Gao, filed Feb. 7, 2001 (hereinafter referred to as "Gao").

As per claim 9, Sun does not expressly disclose ascertaining and identifying type.

However, in an analogous environment, Gao teaches internationalizing software using a semantic analysis phase:

ascertaining a type of code elements using a grammar for the source code (page 3, paragraph 51); *and*

identifying, based on the type of the code elements, any locale-sensitive content

delimited by the code elements (page 3, paragraph 52).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to use Gao's type analysis and identification determination in Sun's

internationalization method. One of ordinary skill would have been motivated to detect

potential internationalization problems using language grammar rules.

11.   Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sun as applied to

claim 7 above, and further in view of Hinks.

As per claim 11, Sun does not expressly disclose storing content in a database file.

However, Hinks teaches the use of a database file for storage of locale sensitive

content (column 3 lines 13-16).

It would have been obvious to one of ordinary skill in the art at the time the

invention was made to store Sun's localization data in Hinks database file. One of

ordinary skill would have been motivated to store localization data in an easily searchable

data structure.

12.   Claims 39 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hinks

as applied to claim 32 above, and further in view of Gao.

As per claim 39, the above rejection of claim 32 is incorporated. Further, Hinks does not expressly disclose a runtime manager.

However, Gao teaches the use of an "Integrated Translation Environment" which populates web documents with locale-sensitive content prior to serving (page 5 paragraph 0135).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Gao's web localizer with Hinks' internationalization method. One of ordinary skill would have been motivated to present localized content in a document.

As per claim 40, the above rejection of claim 32 is incorporated. Hinks further discloses a translated version of the locale-sensitive content (column 3 lines 23-24).

Hinks does not expressly disclose a runtime manager.

However, Gao teaches the use of an "Integrated Translation Environment" which populates web documents with locale-sensitive content during runtime (page 5 paragraph 0135).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Gao's web localizer with Hinks' internationalization method. One of ordinary skill would have been motivated to present localized content in a document when it is presented.
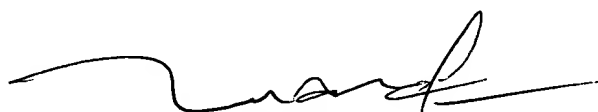
Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (703) 605-5233. The examiner can normally be reached on M-F 6:30-3:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr

TUAN DAM
SUPERVISORY PATENT EXAMINER